



U.S. Department of Energy  
Office of Electricity Delivery  
and Energy Reliability

INL/EXT-10-16381

# NSTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses

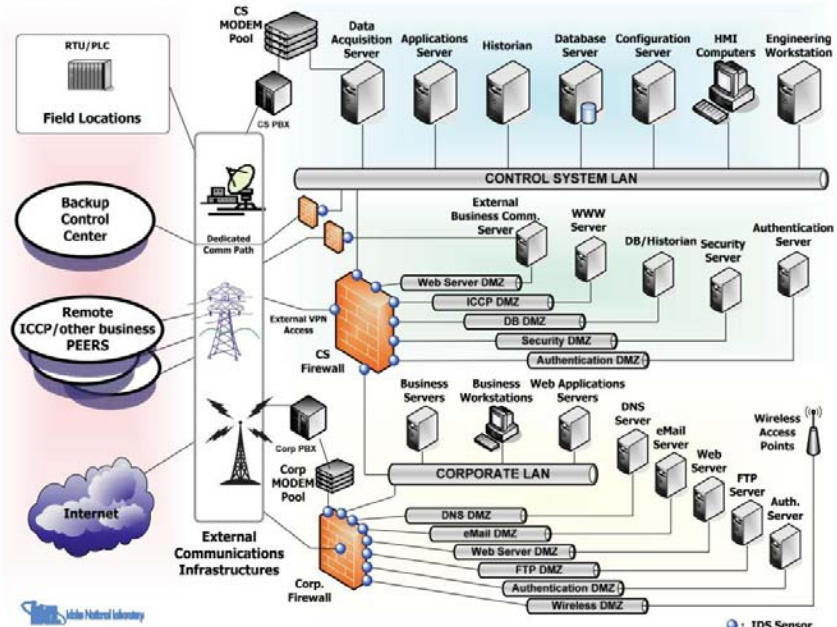
May 2010

## NSTB

National SCADA Test Bed  
Enhancing control systems security in the energy sector



## SECURE CONTROL SYSTEM/ENTERPRISE ARCHITECTURE



# Idaho National Labs SCADA Report

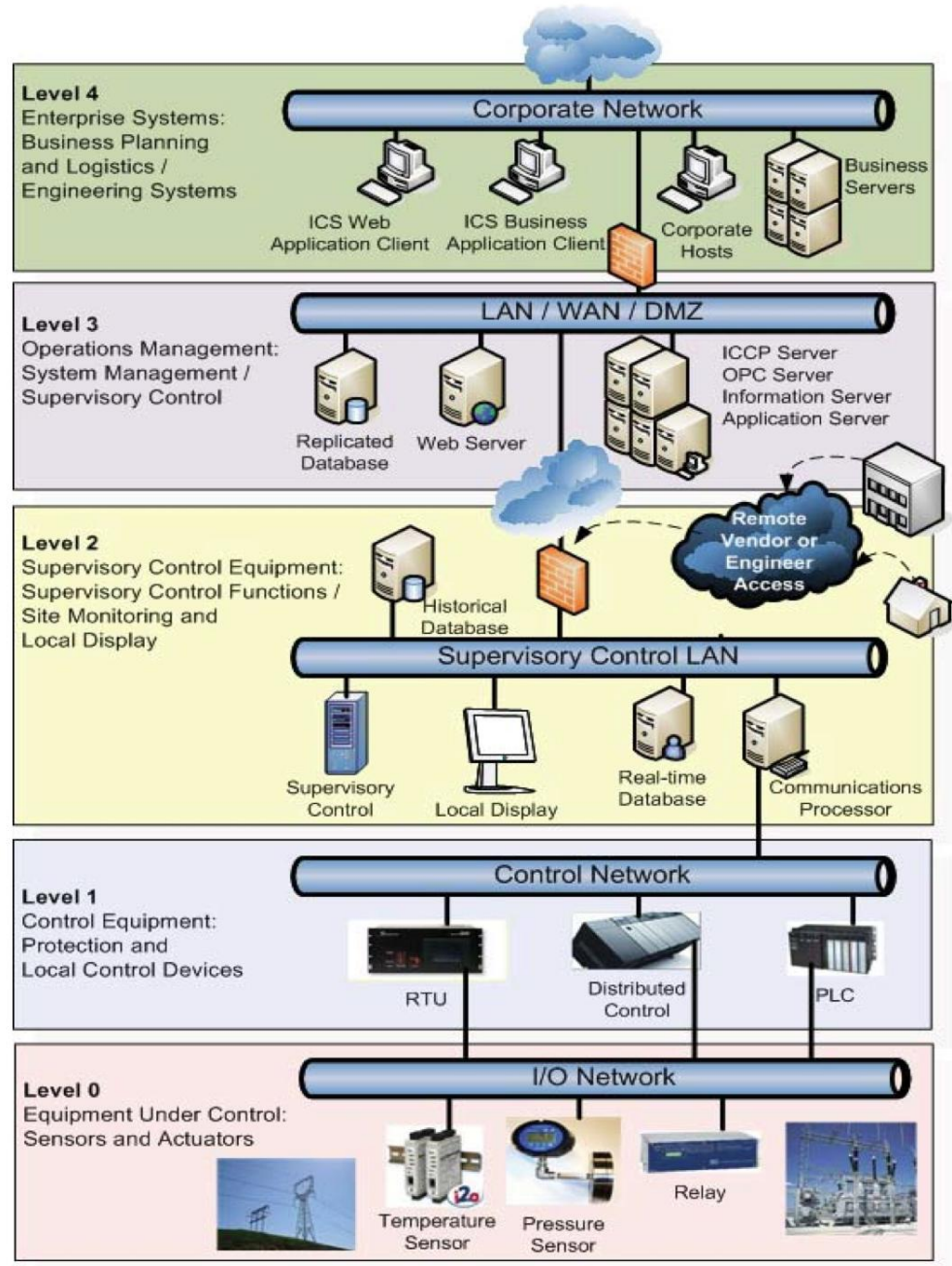


Table 27. Most common programming errors found in ICS code.

<b>Weakness Classification</b>	<b>Vulnerability Type</b>
CWE-19: Data Handling	CWE-228: Improper Handling of Syntactically Invalid Structure
	CWE-229: Improper Handling of Values
	CWE-230: Improper Handling of Missing Values
	CWE-20: Improper Input Validation
	CWE-116: Improper Encoding or Escaping of Output
	CWE-195: Signed to Unsigned Conversion Error
	CWE-198: Use of Incorrect Byte Ordering
CWE-119: Failure to Constrain Operations within the Bounds of a Memory Buffer	CWE-120: Buffer Copy without Checking Size of Input (“Classic Buffer Overflow”)
	CWE-121: Stack-based Buffer Overflow
	CWE-122: Heap-based Buffer Overflow
	CWE-125: Out-of-bounds Read
	CWE-129: Improper Validation of Array Index
	CWE-131: Incorrect Calculation of Buffer Size
	CWE-170: Improper Null Termination
	CWE-190: Integer Overflow or Wraparound
	CWE-680: Integer Overflow to Buffer Overflow
CWE-398: Indicator of Poor Code Quality	CWE-454: External Initialization of Trusted Variables or Data Stores
	CWE-456: Missing Initialization
	CWE-457: Use of Uninitialized Variable
	CWE-476: NULL Pointer Dereference
	CWE-400: Uncontrolled Resource Consumption (“Resource Exhaustion”)
	CWE-252: Unchecked Return Value
	CWE-690: Unchecked Return Value to NULL Pointer Dereference
	CWE-772: Missing Release of Resource after Effective Lifetime
CWE-442: Web Problems	CWE-22: Improper Limitation of a Pathname to a Restricted Directory (“Path Traversal”)
	CWE-79: Failure to Preserve Web Page Structure (“Cross-site Scripting”)
	CWE-89: Failure to Preserve SQL Query Structure (“SQL Injection”)
CWE-703: Failure to Handle Exceptional Conditions	CWE-431: Missing Handler
	CWE-248: Uncaught Exception
	CWE-755: Improper Handling of Exceptional Conditions
	CWE-390: Detection of Error Condition Without Action



# Linkage with Fundamental Changes in Enterprise Security Initiatives

## Twenty Critical Controls for Effective Cyber Defense Guidelines

What the 20 CSC Critics say...

### 20 Critical Security Controls - Version 2.0

- 20 Critical Security Controls - Introduction (Version 2.0)
- Critical Control 1: Inventory of Authorized and Unauthorized Devices
- Critical Control 2: Inventory of Authorized and Unauthorized Applications
- Critical Control 3: Secure Configurations for Hardware and Software
- Critical Control 4: Secure Configurations for Network Devices
- Critical Control 5: Boundary Defense
- Critical Control 6: Maintenance, Monitoring, and Analysis of Systems
- **Critical Control 7: Application Software Security**
- Critical Control 8: Controlled Use of Administrative Privileges
- Critical Control 9: Controlled Access Based on Need to Know
- Critical Control 10: Data Protection
- Critical Control 11: Incident Response and Computer Forensics
- Critical Control 12: Penetration Testing
- Critical Control 13: Security Awareness and Training
- Critical Control 14: Vendor Management
- Critical Control 15: Physical Security
- Critical Control 16: Telecommunications and Network Security
- Critical Control 17: Wireless Network Security
- Critical Control 18: Cloud Security
- Critical Control 19: Mobile Device Security
- Critical Control 20: Security of Information Systems

## CAG: Critical Control 7: Application Software Security

<< previous control

Consensus Audit Guidelines

next control >>

### How do attackers exploit the lack of this control?

Attacks against vulnerabilities in web-based and other application software have been a top priority for criminal organizations in recent years. Application software that does not properly check the size of user input, fails to sanitize user input by filtering out unneeded but potentially malicious character sequences, or does not initialize and clear variables properly could be vulnerable to remote compromise. Attackers can inject specific exploits, including buffer overflows, SQL injection attacks, and cross-site scripting code to gain control over vulnerable machines. In one attack in 2008, more than 1 million web servers were exploited and turned into infection engines for visitors to those sites using SQL injection. During that attack, trusted websites from state governments and other organizations compromised by attackers were used to infect hundreds of thousands of

## CWE and CAPEC included in Control 7 of the “Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines”

### Procedures and tools for implementing this control

Source code testing tools, web application security scanning tools, and object code testing tools have proven useful in securing application software, along with manual application security penetration testing by testers who have extensive programming knowledge as well as application penetration testing expertise. The Common Weakness Enumeration (CWE) initiative is utilized by many such tools to identify the weaknesses that they find. Organizations can also use CWE to determine which types of weaknesses they are most interested in addressing and removing. A broad community effort to identify the “Top 25 Most Dangerous Programming Errors” is also available as a minimum set of important issues to investigate and address during the application development process. When evaluating the effectiveness of testing for these weaknesses, the Common Attack Pattern Enumeration and Classification (CAPEC) can be used to organize and record the breadth of the testing for the CWEs as well as a way for testers to think like attackers in their development of test cases.





ISO/IEC JTC 1/SC 27 Nxxxx

ISO/IEC JTC 1/SC 27/WG x Nxxxxx

REPLACES: N

ISO/IEC JTC 1/SC 27

Information technology - Security techniques  
Secretariat: DIN, Germany

**DOC TYPE:** NB NWI Proposal for a technical report (TR)

**TITLE:** National Body New Work Item Proposal on "Secure software development and evaluation under ISO/IEC 15408 and ISO/IEC 18405"

**SOURCE:** INCITS/CS1, National Body of (US)

**DATE:** 2009-09-30

**PROJECT:** 15408 and 18405

**STATUS:** This document is circulated for consideration at the forthcoming meeting of SC 27/WG 3 to be held in Redmond (WA, USA) on 2<sup>nd</sup> – 6<sup>th</sup> November 2009.

**ACTION ID:** ACT

**DUE DATE:**

**DISTRIBUTION:** P, O- and L-Members  
W. Fumy, SC 27 Chairman  
M. De Soete, SC 27 Vice-Chair  
E. J. Humphreys, K. Naemura, M. Bañón, M.-C. Kang, K. Rannenberg, WG-Conveners

**MEDIUM:** Livelin-server

**NO. OF PAGES:** xx

## Common Criteria v4 CCDB

- TOE to leverage CAPEC & CWE
- Also investigating how to leverage ISO/IEC 15026

## NIAP Evaluation Scheme

- Above plus
- Also investigating how to leverage SCAP

### New Work Item Proposal

#### NP submitting

#### PROPOSAL FOR A NEW WORK ITEM

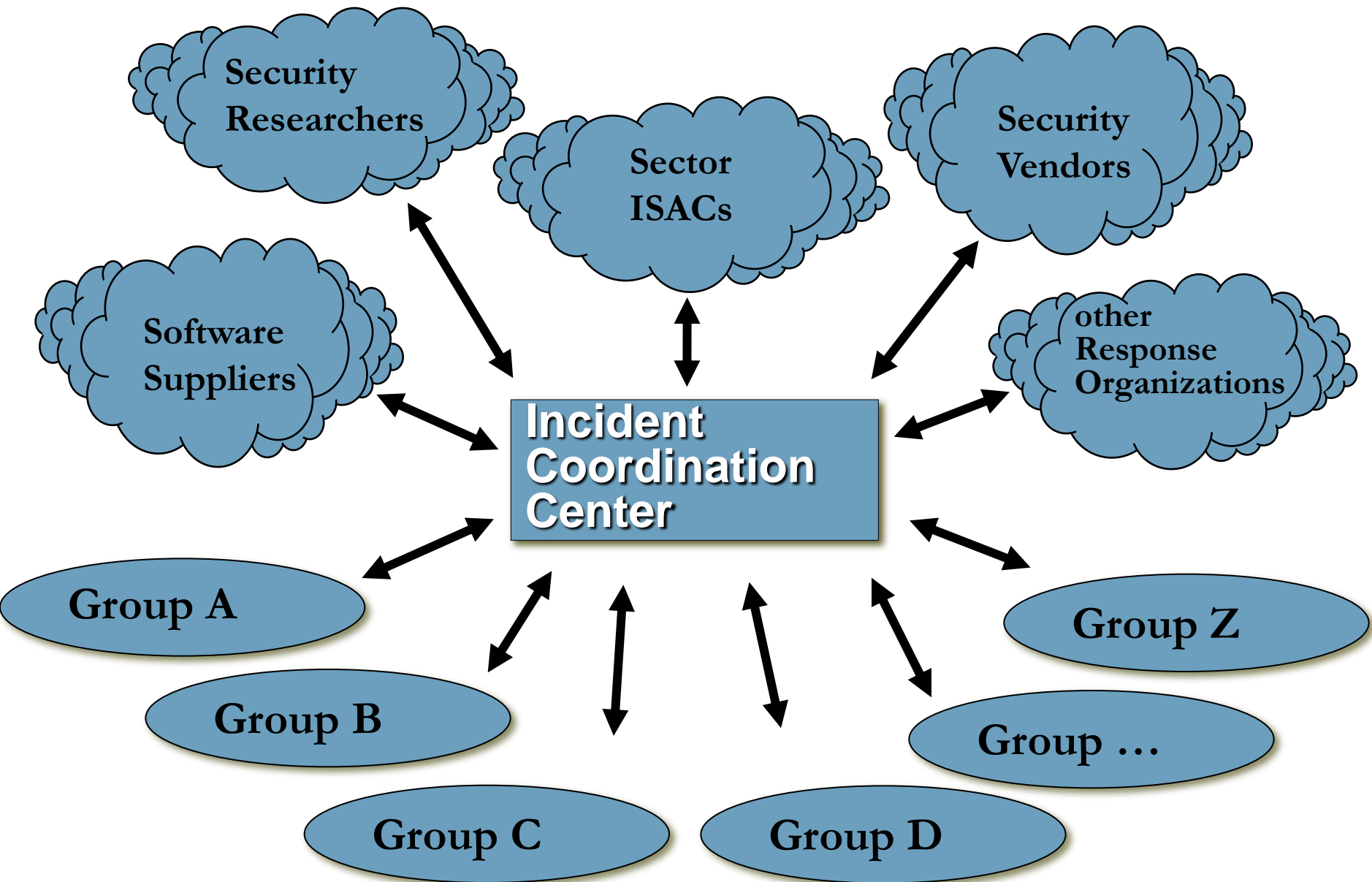
Date of presentation of proposal: YYYY-MM-DD	Proposer: ISO/IEC JTC 1 SC27
Secretariat: National Body	ISO/IEC JTC 1 N XXXX ISO/IEC JTC 1/SC 27 N

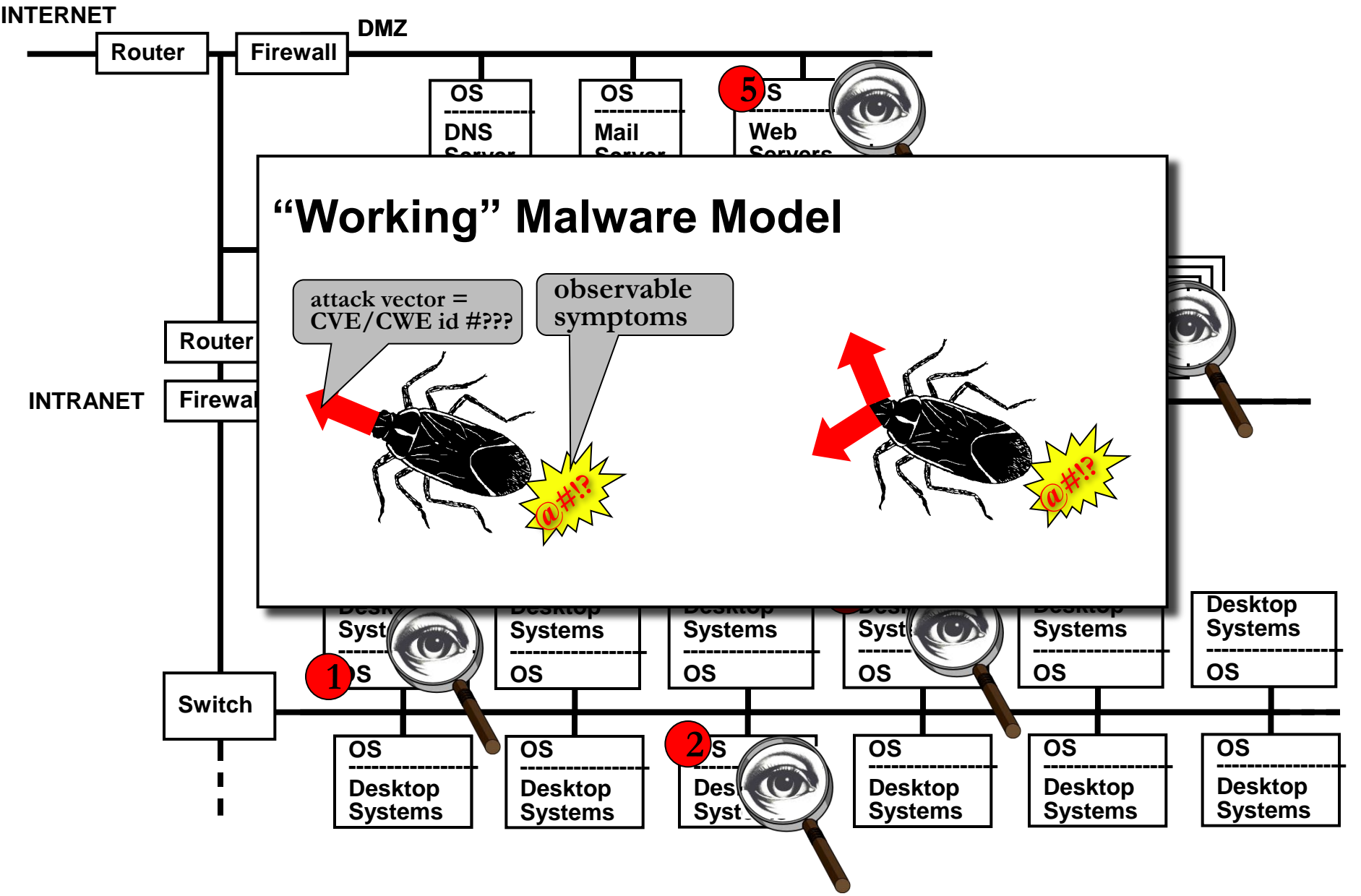
A proposal for a new work item shall be submitted to the secretariat of the ISO/IEC joint technical committee concerned with a copy to the ISO Central Secretariat.

#### Presentation of the proposal

<b>Title</b> Secure software development and evaluation under ISO/IEC 15408 and ISO/IEC 18405
<b>Scope</b> In the case where a target of evaluation (TOE) being evaluated, under ISO/IEC 15408 and ISO/IEC 18405, includes specific software portions, the TOE developer may optionally present the developer's technical rationale for mitigating software common attack patterns and related weaknesses as described in the latest revision of the Common Attack Pattern Enumeration and Classification (CAPEC) available from <a href="http://capec.mitre.org/">http://capec.mitre.org/</a> . The developer's technical rationale is expected to include a range of mitigation techniques, from architectural properties to design features, coding techniques, use of tools or other means. This Technical Report (TR) provides guidance for the developer and the evaluator on how to use the CAPEC as a technical reference point during the TOE development life cycle and in an evaluation of the TOE secure software under ISO/IEC 15408 and 18045, by addressing: a) A refinement of the IS 15408 Attack Potential calculation table for software, taking into account the entries contained in the CAPEC and their characterization. b) How the information for mitigating software common attack patterns and related weaknesses is used in an IS 15408 evaluation, in particular providing guidance on how to determine which attack patterns and weaknesses are applicable to the TOE, taking into consideration of 1. the TOE technology; 2. the TOE security problem definition; 3. the interfaces the TOE exports that can be used by potential attackers; 4. the Attack Potential that the TOE needs to provide resistance for. c) How the technical rationale provided by the developer for mitigating software common attack patterns and related weaknesses is used in the evaluation of the TOE design and the development of test cases. d) How the CAPEC and related Common Weakness Enumeration (CWE) taxonomies are used by the evaluator, who needs to consider all the applicable attack patterns and be able to exploit specific related software weaknesses while performing the subsequent vulnerability analysis (AVA_VAN) activities on the TOE. e) How incomplete entries from the CAPEC are resolved during an IS 15408 evaluation. f) How the evaluator's attack and weakness analysis of the TOE incorporates other attacks and weaknesses not yet documented in the CAPEC. The TR also investigates specific elements from the ISO/IEC 15026 (and its revision) are applicable to the guidelines being developed in the TR within the context of IS 15408 and 18405.



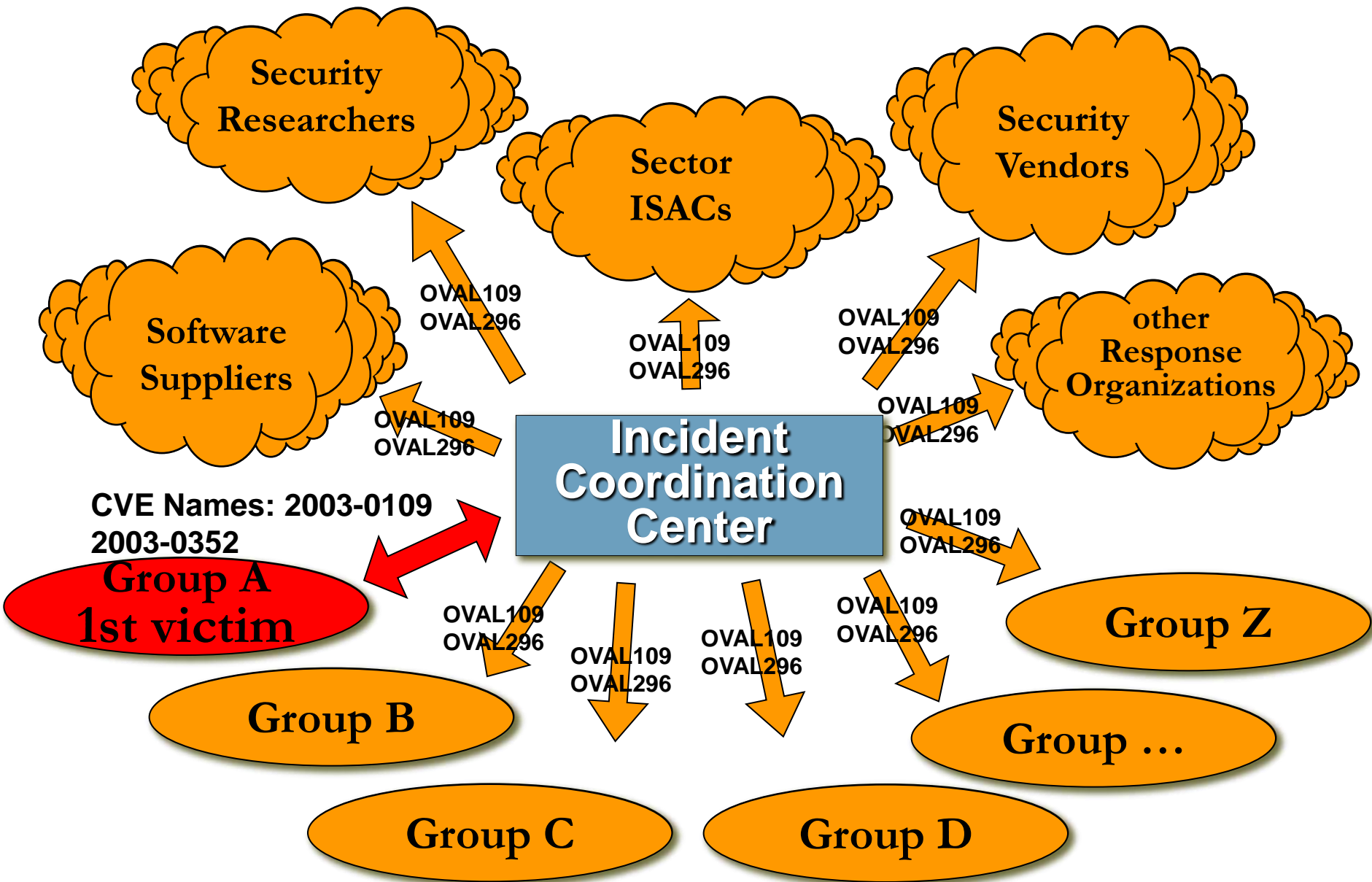




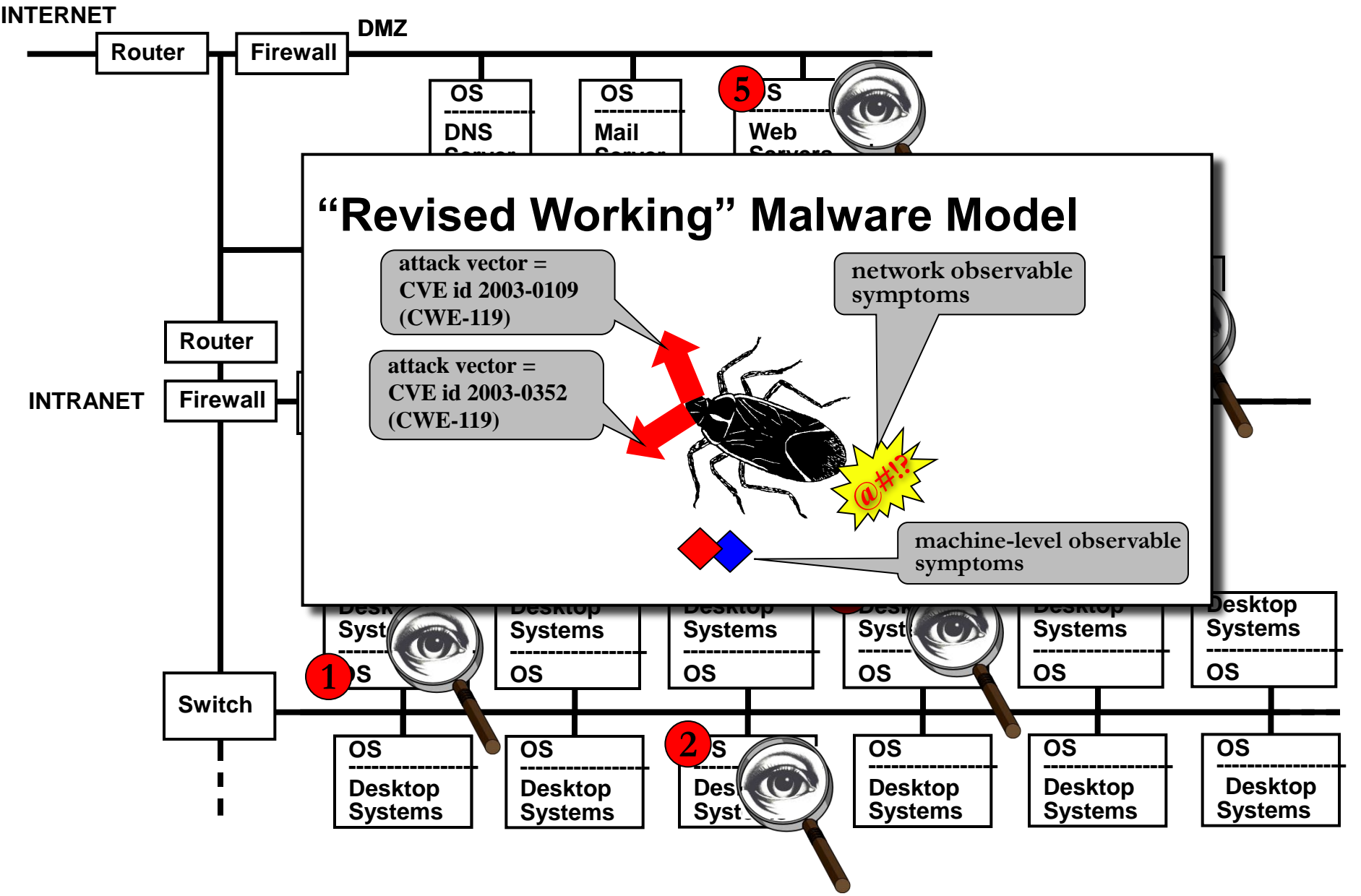
“Group A” Network

# First Level Vulnerability Examination Results

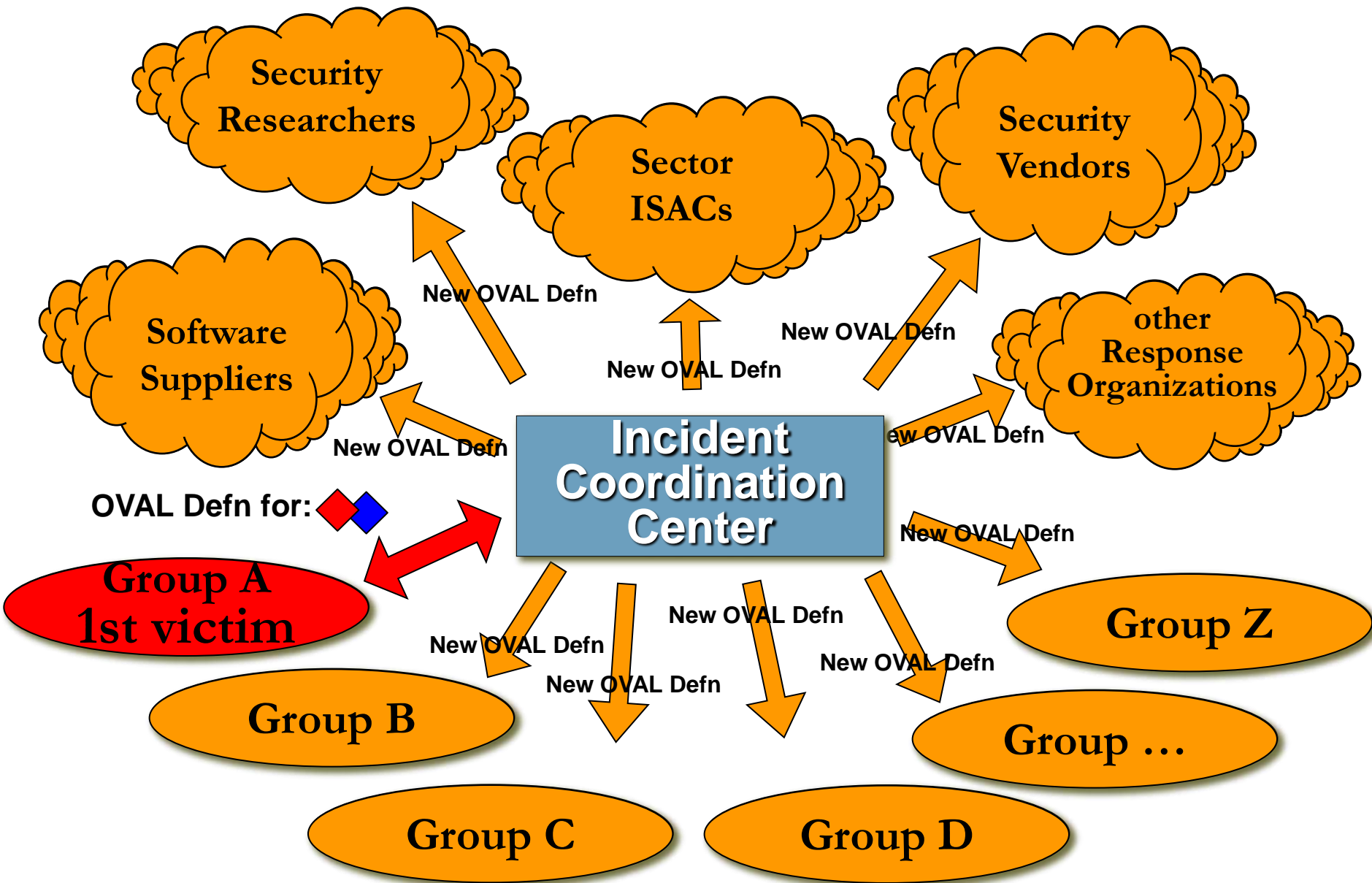
	CVE Name: 2003-0109 <b>OVAL109</b>	CVE Name: 2003-0352 <b>OVAL296</b>	CVE Name: 2003-0223 <b>OVAL66</b>	CVE Name: 2003-0228 <b>OVAL321</b>	CVE Name: 2003-0660 <b>OVAL198</b>
System 1 10.0.0.121	no	yes	no	yes	yes
System 2 10.0.0.122	no	yes	no	no	no
System 3 10.0.0.123	no	yes	no	yes	no
System 4 10.0.1.124	yes	no	yes	no	yes
System 5 10.0.2.125	yes	no	no	no	no



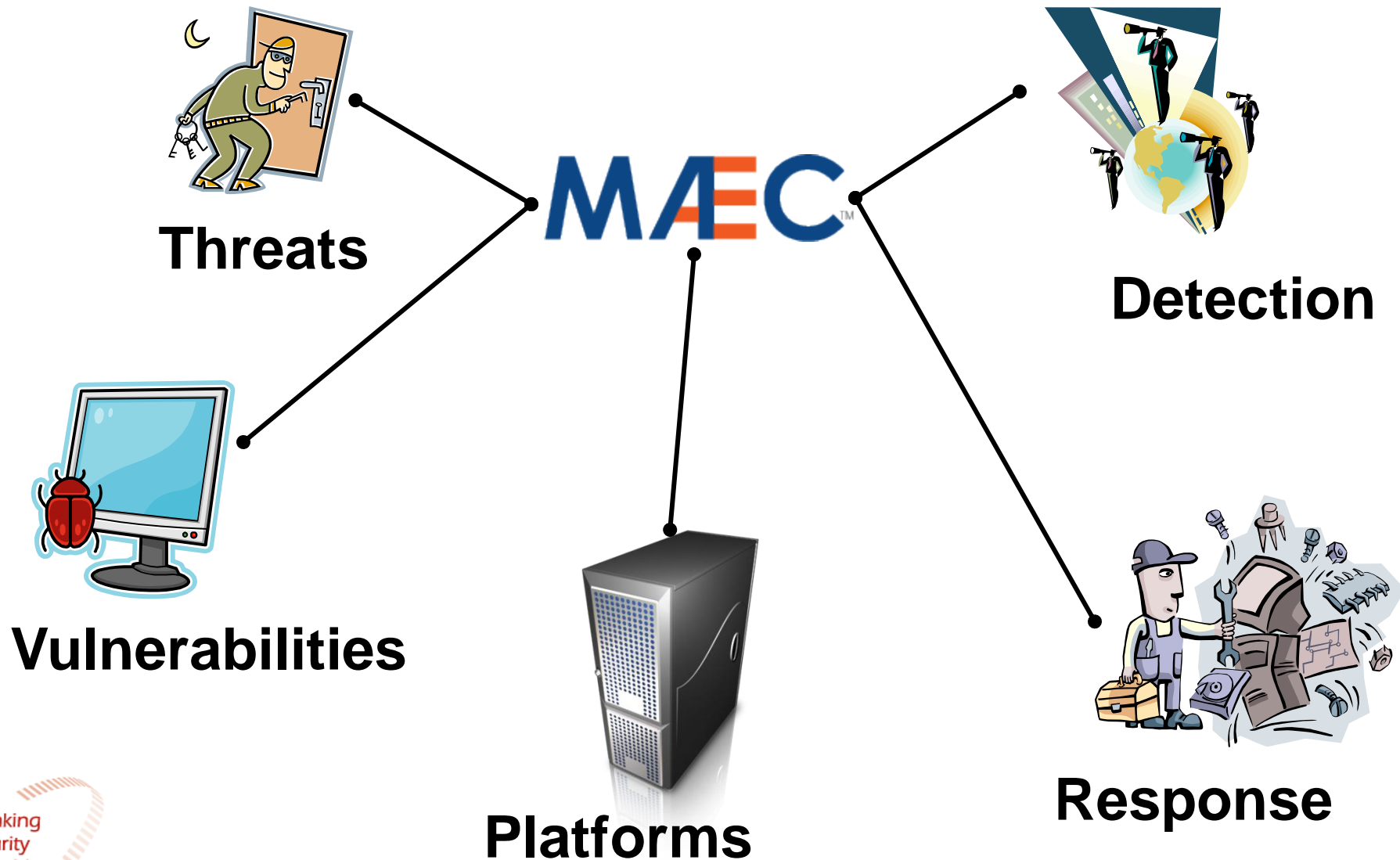




“Group A” Network

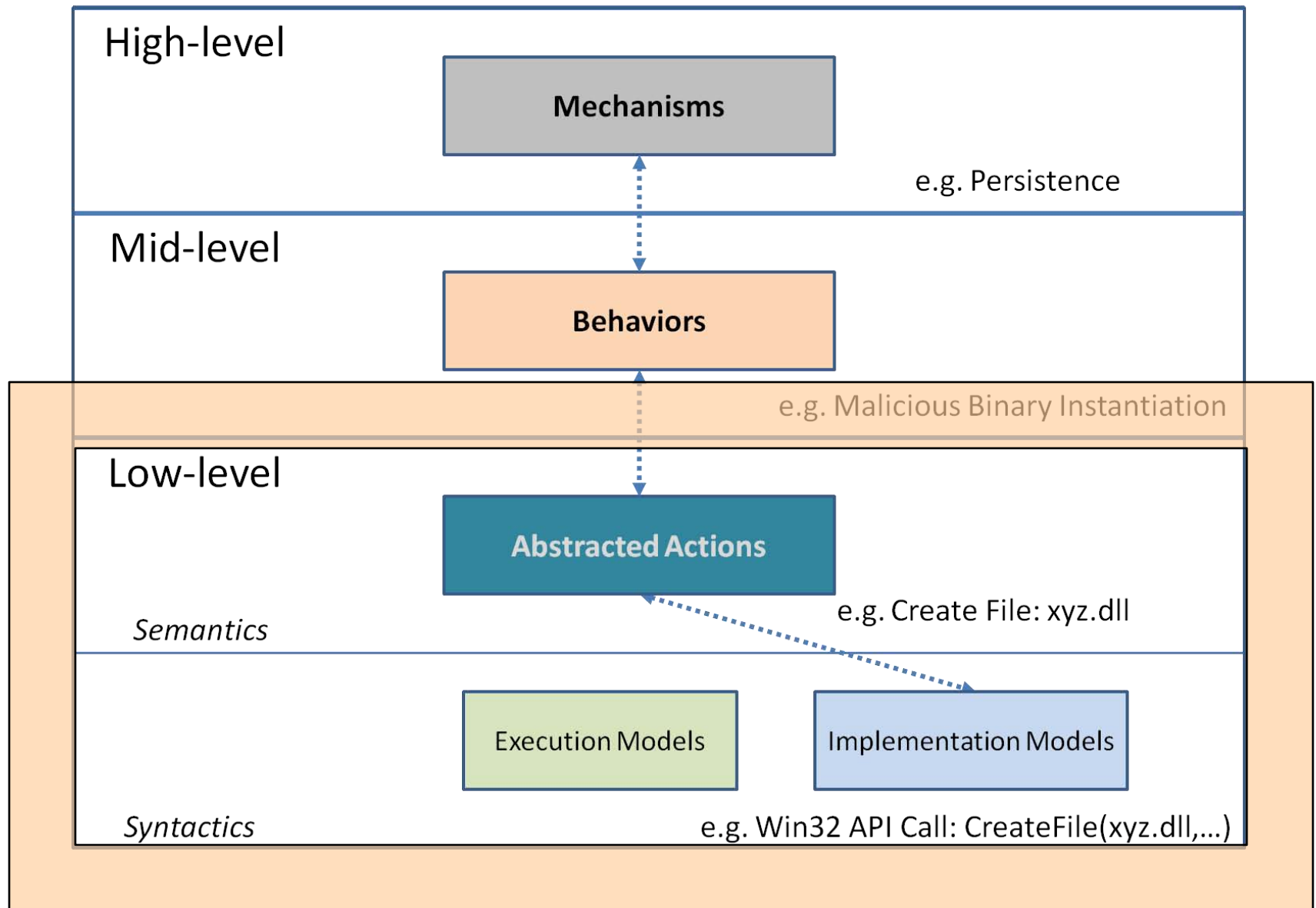


# Correlate, Integrate, Automate

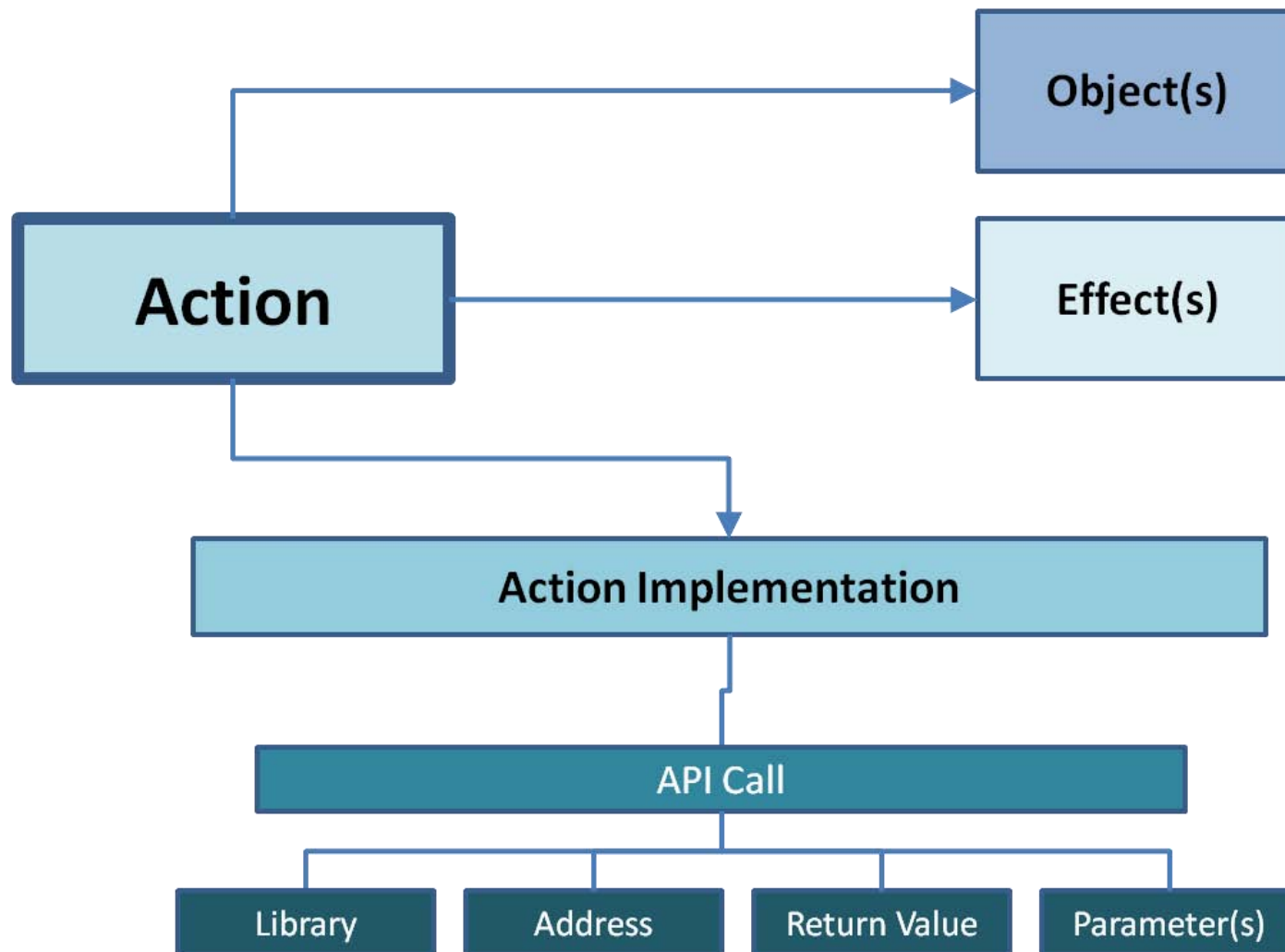




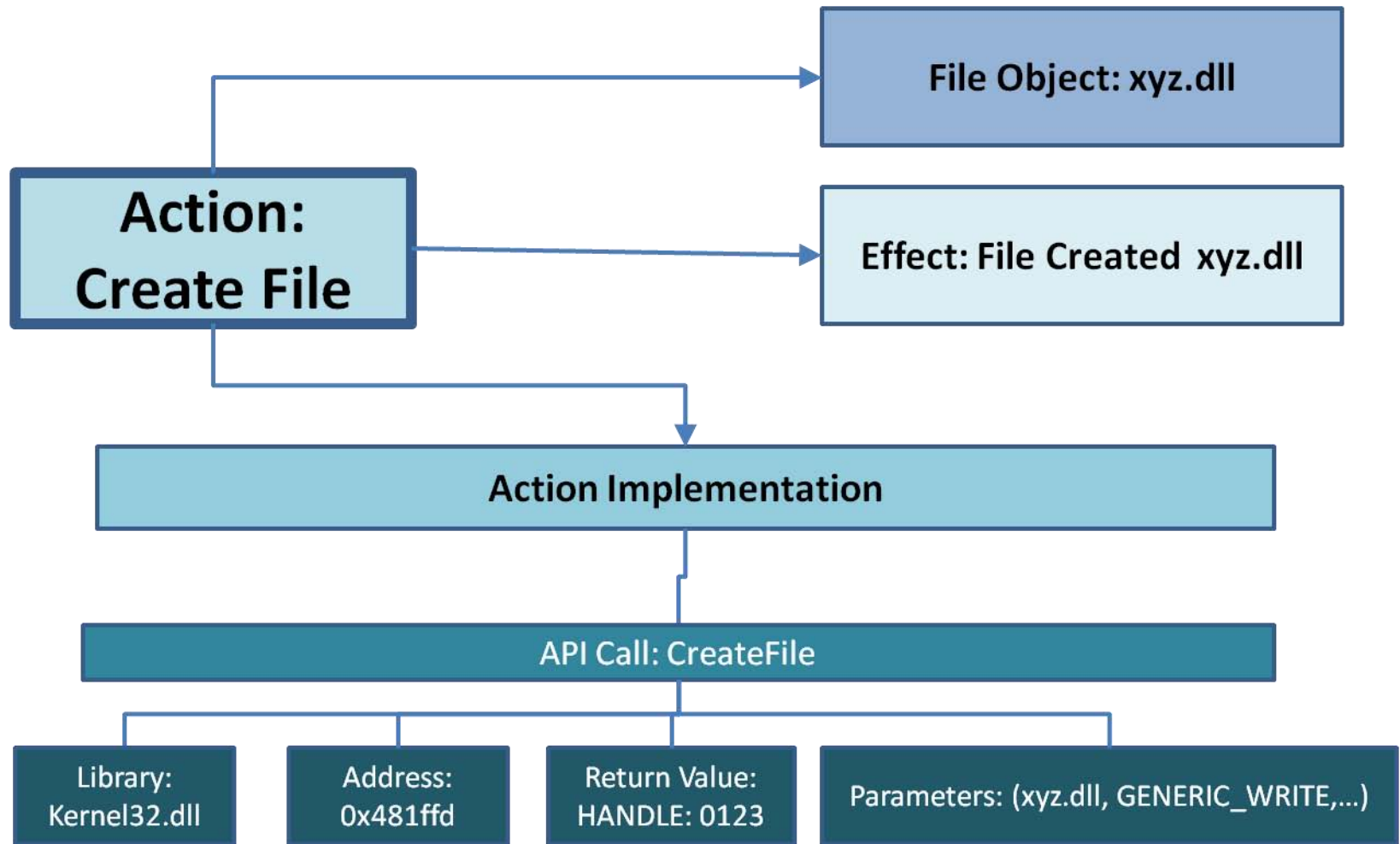
# High-level MAEC Overview



# MAEC Action Model

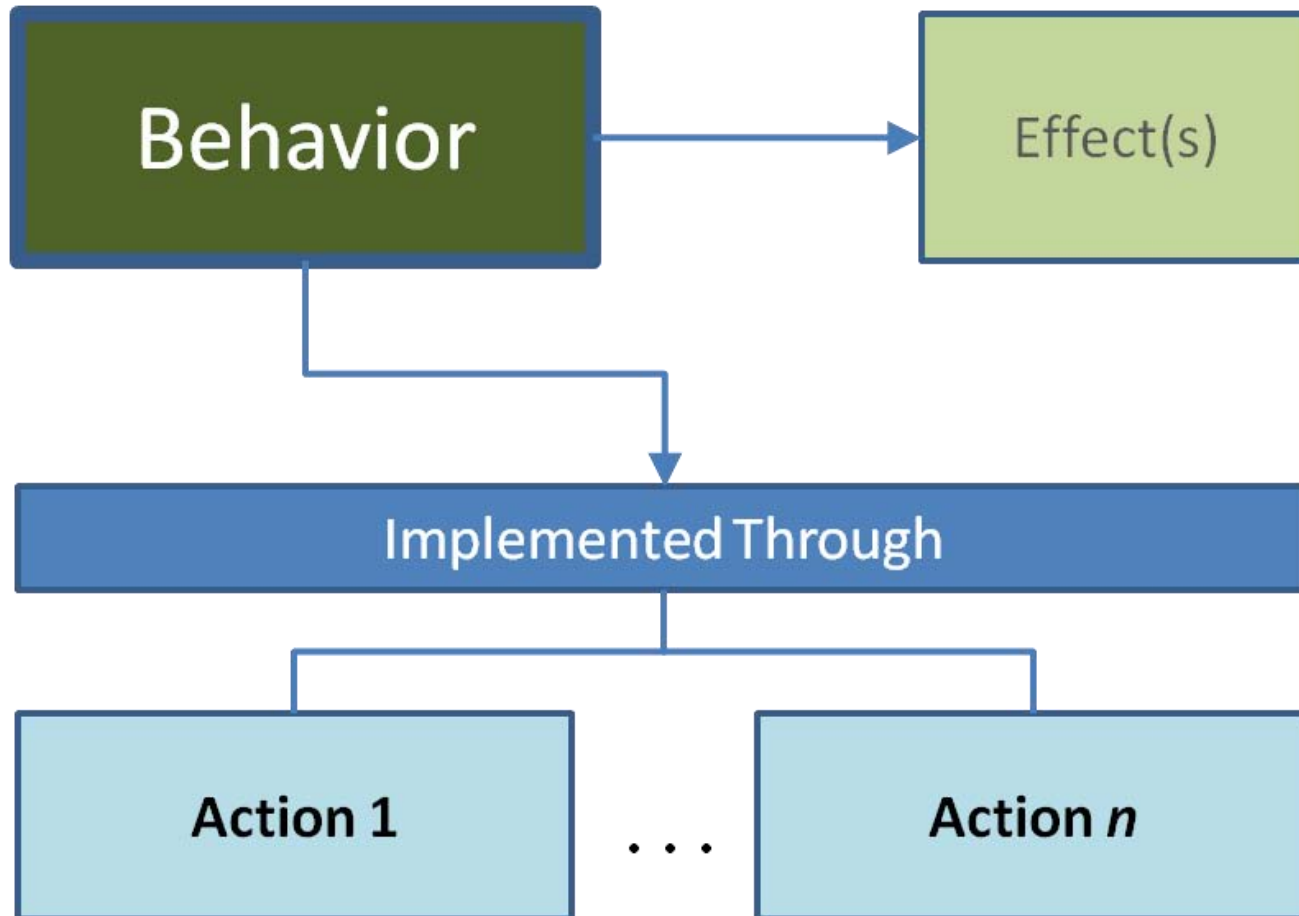


# Action Example

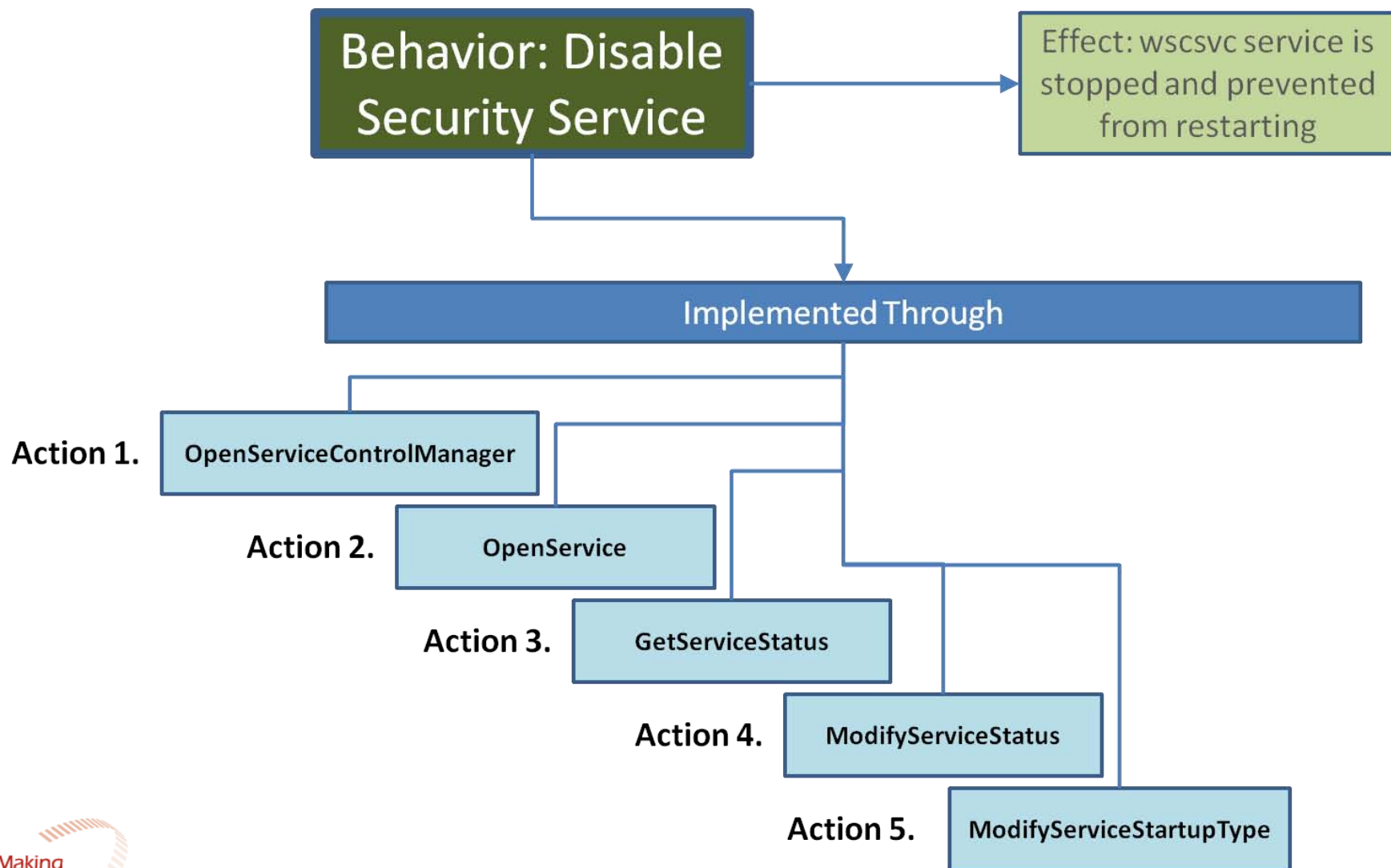




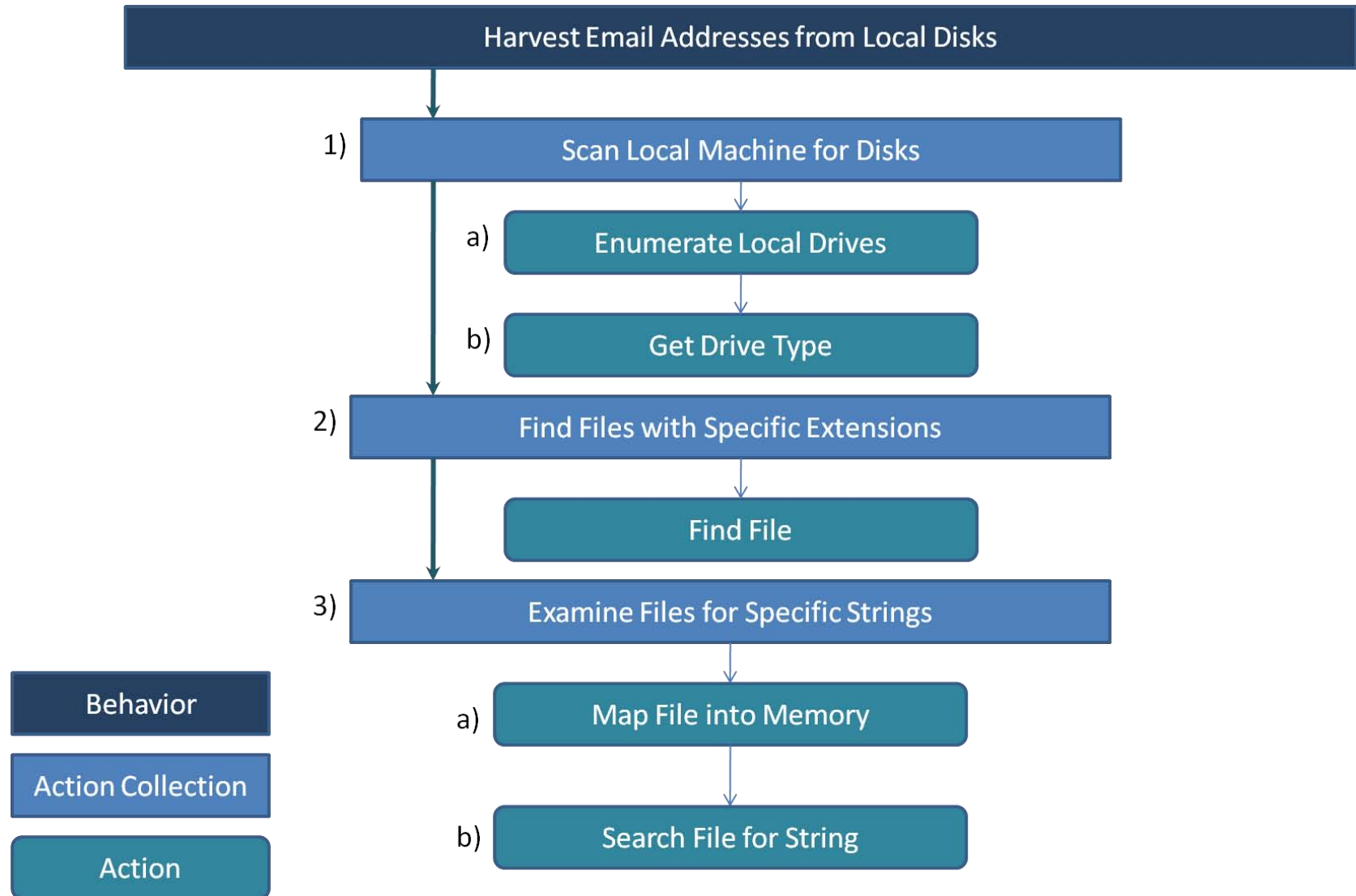
# MAEC Behavior Model



# Basic Behavior Example



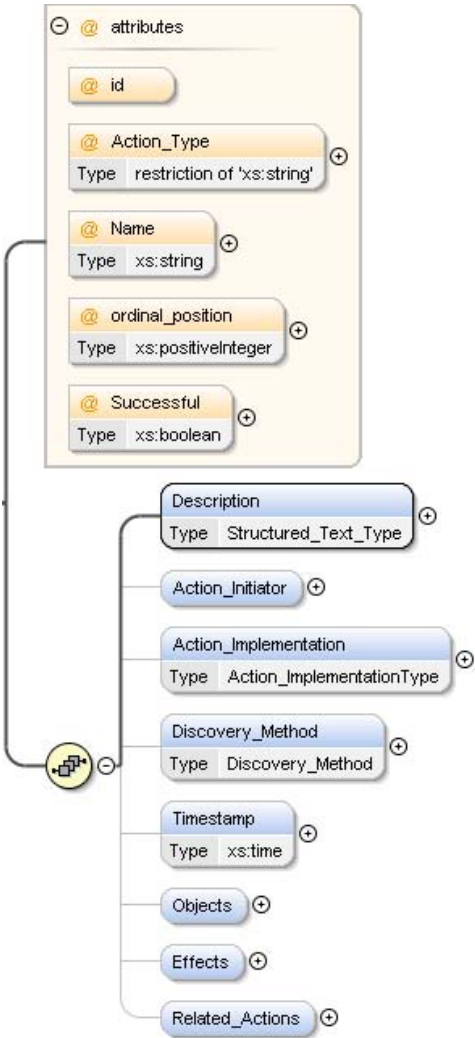
# More Complex Behavior Example



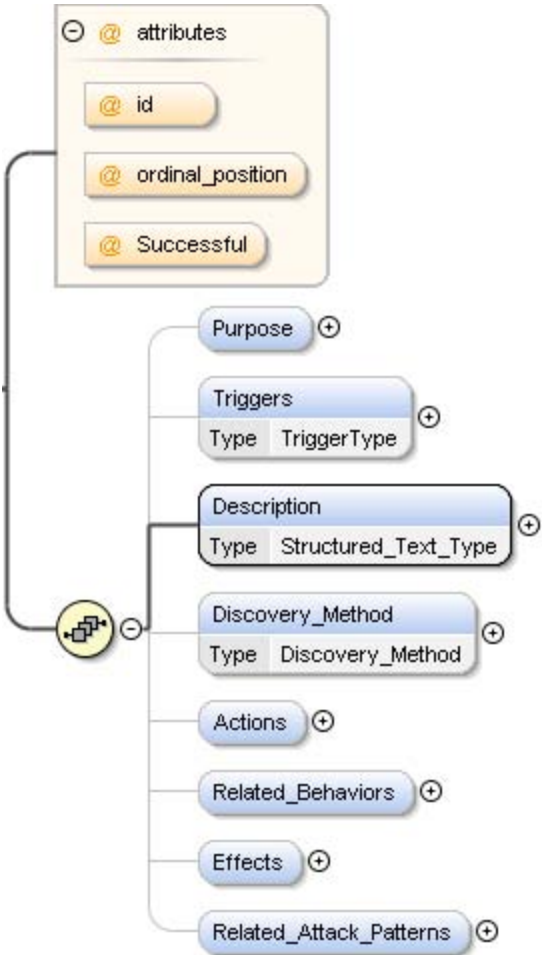


# MAEC Schema Overview – Initial Release

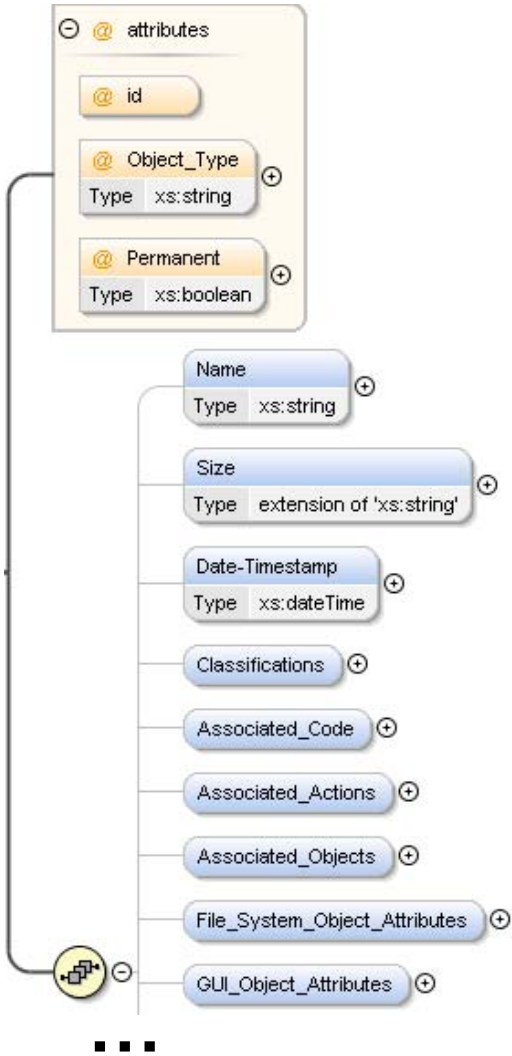
## ActionType



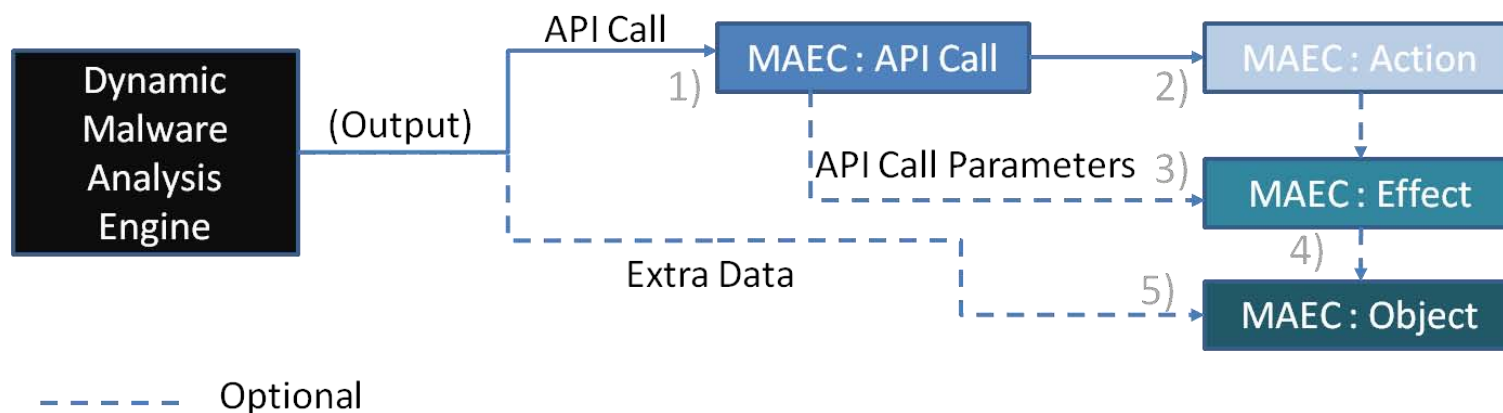
## BehaviorType



## ObjectType



# Dynamic Malware Analysis <-> MAEC



## Process

- 1) An API call is captured by the analysis engine and mapped to MAEC's enumeration of API calls.
- 2) The MAEC enumerated call is mapped to its corresponding action.
- 3) The MAEC defined action is mapped to a corresponding MAEC effect (as necessary), which is populated by the parameters of the call.
- 4) The MAEC effect is linked to a MAEC object (as necessary).
- 5) Any extra data output (e.g. file attributes, network capture, etc.) from the analysis engine is mapped to its corresponding object (as necessary).

# Test Case: CWSandbox Output -> MAEC

```
PID:1080,TID:1812,Caller:$00400000("KB823988.exe"),BEFORE,typFileSystem."FindFirstFile"
PID:1080,TID:1812,Caller:$00400000("KB823988.exe"),BEFORE,typFileSystem."SetFileAttrib"
PID:1080,TID:1812,Caller:$00400000("KB823988.exe"),BEFORE,typFileSystem."DeleteFileW"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumKeyA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumKeyA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumValueW"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegEnumValueW"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExA"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExW"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegOpenKeyExW"
PID:1080,TID:1812,Caller:$77A80000("CRYPT32.dll"),AFTER,typRegistry."RegCreateKeyExW"
```

```
<Action Successful="true" id="10" Action_Type="copy" Name="copy_file">
  <Description/>
  <Action_Initiator type="Process">
    <Initiator_Name>KB823988.exe</Initiator_Name>
    <Process_ID>1080</Process_ID>
    <Thread_ID>1812</Thread_ID>
  </Action_Initiator>
  <Action_Implementation>
    <API_Call>
      <Name>CopyFileW</Name>
      <API_Call_Parameter ordinal_position="1">
        <Name>filetype</Name>
        <Value>file</Value>
      </API_Call_Parameter>
      <API_Call_Parameter ordinal_position="2">
        <Name>srcfile</Name>
        <Value>c:\\KB823988.exe</Value>
      </API_Call_Parameter>
      <API_Call_Parameter ordinal_position="3">
        <Name>dstfile</Name>
        <Value>C:\\WINDOWS\\system32\\ntos.exe</Value>
      </API_Call_Parameter>
      <API_Call_Parameter ordinal_position="4">
        <Name>creationdistribution</Name>
        <Value>CREATE_ALWAYS</Value>
      </API_Call_Parameter>
      <API_Call_Parameter ordinal_position="5">
        <Name>desiredaccess</Name>
        <Value>FILE_ANY_ACCESS</Value>
      </API_Call_Parameter>
      <API_Call_Parameter ordinal_position="6">
        <Name>flags</Name>
        <Value>SECURITY_ANONYMOUS</Value>
      </API_Call_Parameter>
    </API_Call>
  </Action_Implementation>
</Action>
```

## Raw CWSandbox Output

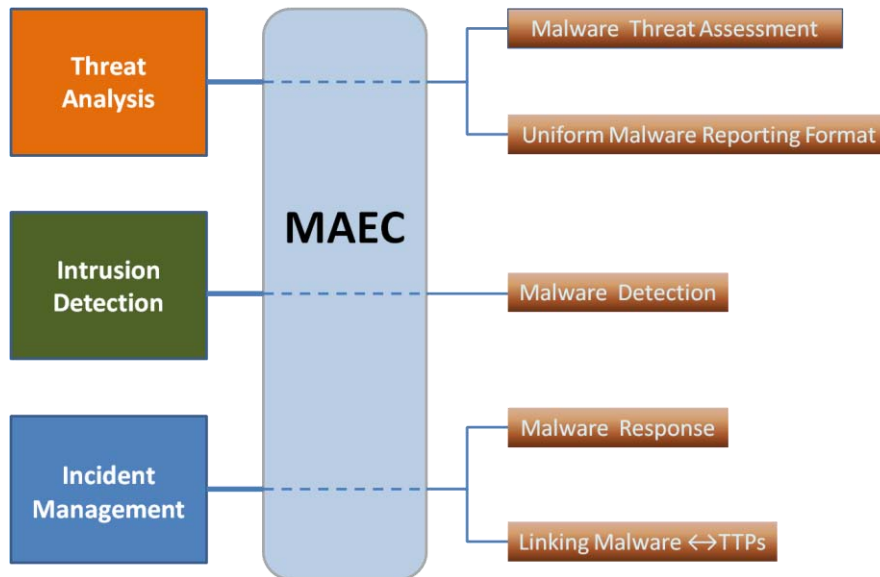
Python  
XSD  
Bindings

MAEC  
XSD

MAEC XML

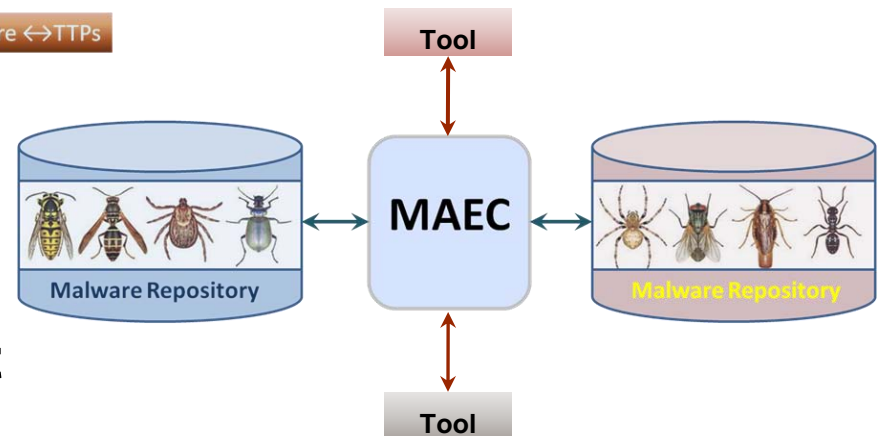
# MAEC Use Cases

- Operational



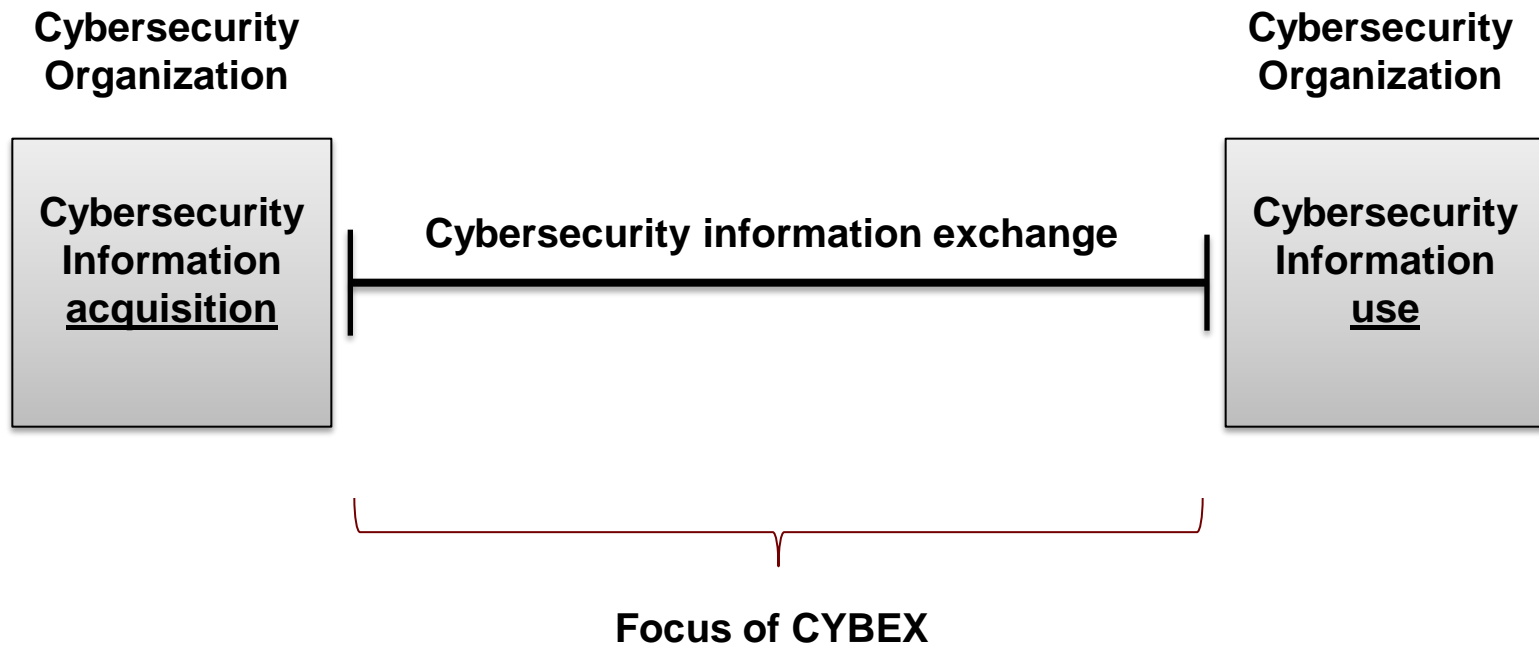
- Analysis

- **Help Guide Analysis Process**
- **Standardized Tool Output**
- **Malware Repositories**



# International Telecommunications Union (ITU) Cyber Security Working Group is creating an exchange standard...

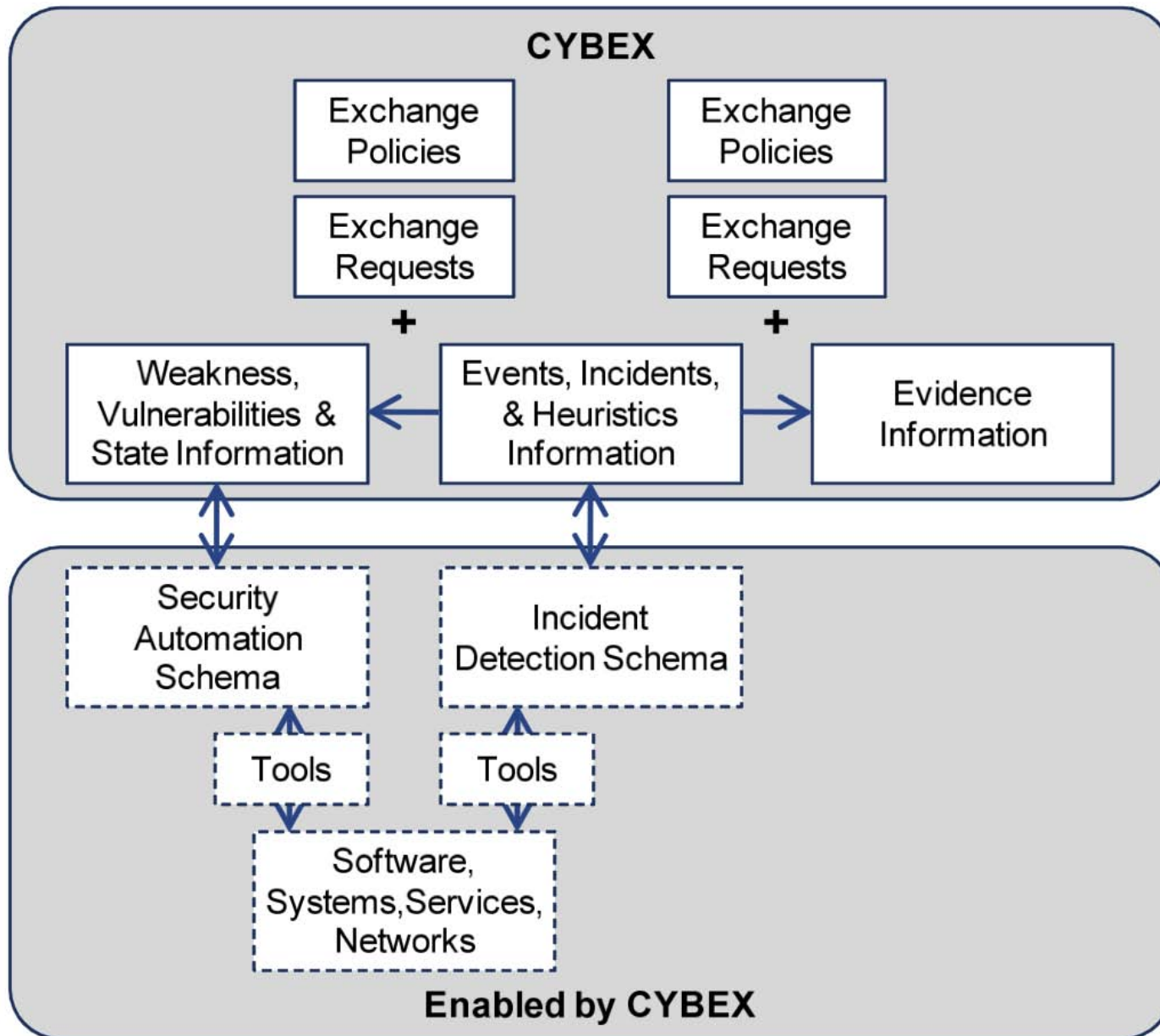
Focus of CYBEX

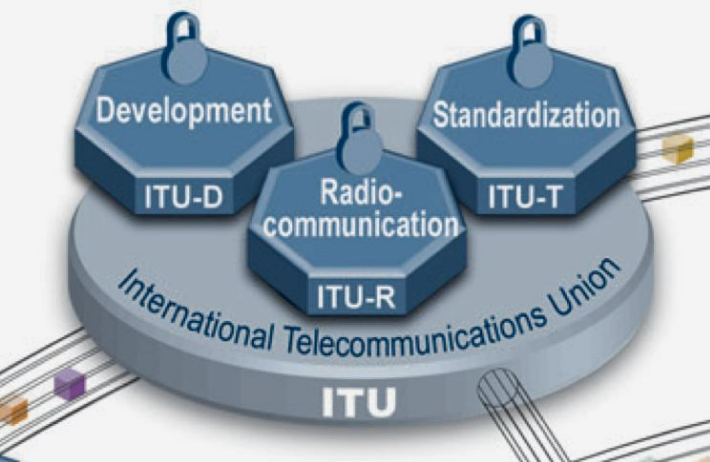


**CYBEX focuses on cybersecurity information exchange between cybersecurity organizations**



# ITU-T CYBersecurity EXchange Framework (CYBEX)





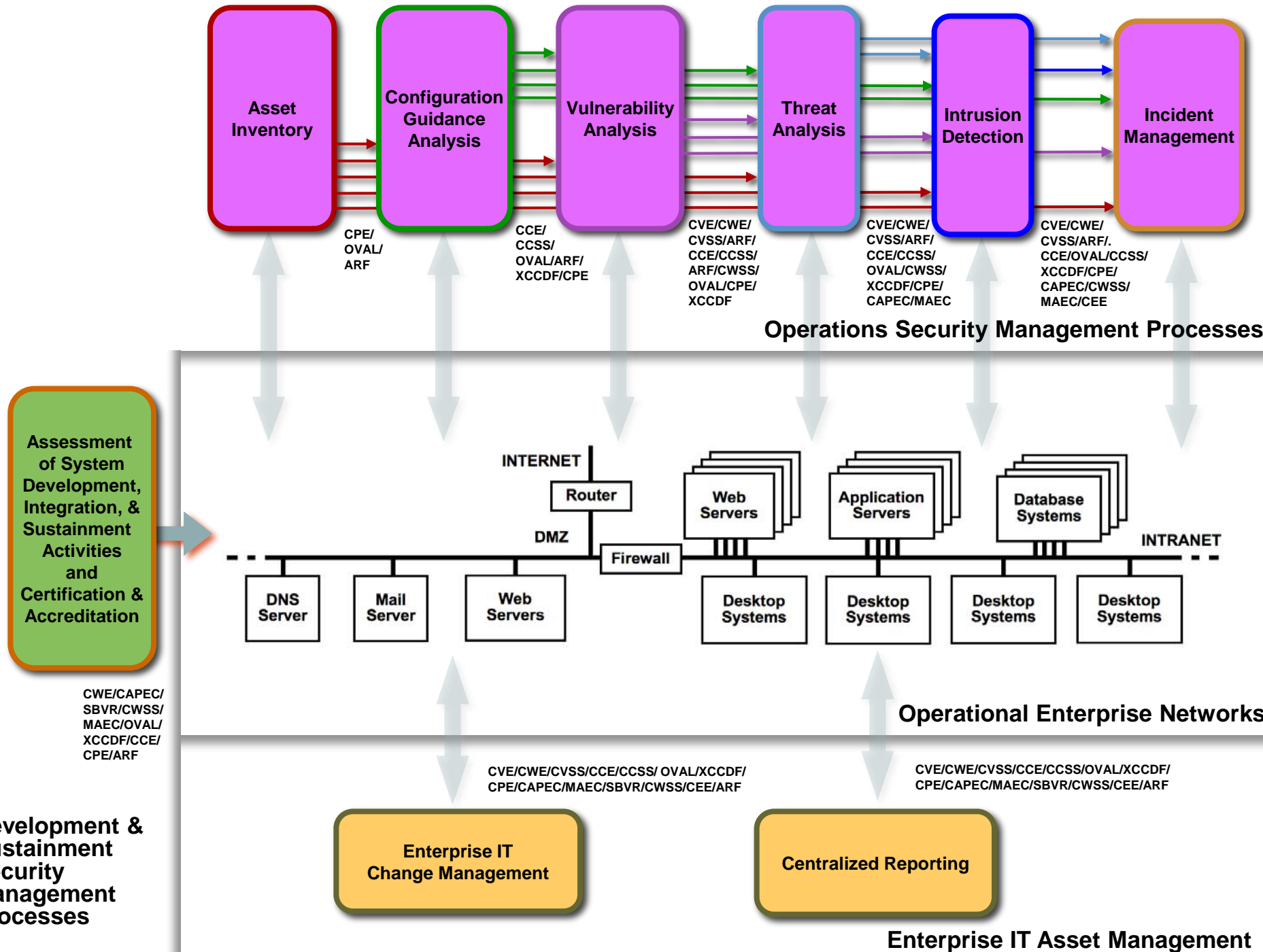
## ITU-T Study Group 17 Question 4 – Cyber Security Cyber Security Exchange Framework (CYBEX)

Creating x.series standards to capture the correct and supported USE of the enumerated concepts and languages – effort stewardship and definition stays with originating organizations

<u>Identifier</u>	<u>Title</u>	<u>Current Text</u>
X.cybief	Cybersecurity Information Exchange Framework	TD406
X.cybief.1	Guidelines for Administering the OID arc for cybersecurity information exchange	TD406
<b>X.cce</b>	<b>Common Configuration Enumeration</b>	TD406
<b>X.cce</b>	<b>Common Event Expression</b>	TD406
X.chirp	Cybersecurity Heuristics and Information Request Protocol	TD406
<b>X.cpe</b>	<b>Common Platform Enumeration</b>	TD406
<b>X.crf</b>	<b>Common Result Format</b>	TD406
<b>X.cve</b>	<b>Common Vulnerabilities and Exposures</b>	TD405
<b>X.cvss</b>	<b>Common vulnerability scoring system</b>	TD412
<b>X.cwe</b>	<b>Common Weakness Enumeration</b>	TD406
<b>X.cwss</b>	<b>Common Weakness Scoring System</b>	TD406
X.dexf	Digital evidence exchange file format	C97
X.dpi	Deep Packet Inspection Exchange Format	TD406
X.gridf	SmartGrid Incident Exchange Format	TD406
<b>X.oval</b>	<b>Open Vulnerability and Assessment Language</b>	TD406
X.pfoc	Phishing, Fraud, and Other Crimeware Exchange Format	TD406
<b>X.scap</b>	<b>Security Content Automation Protocol</b>	TD406
X.teef	Cyber attack tracing event exchange format	C135, C129
<b>X.xccdf</b>	<b>eXensible Configuration Checklist Description Format</b>	TD406
X.cybief-[namespace],	Cybersecurity Information Exchange Namespace	C148
X.cybief-discovery	Cybersecurity Information Exchange Discovery	C145
<b>X.capec</b>	<b>Common Attack Pattern Enumeration and Classification</b>	TD406
X.iodef	Incident Object Description Exchange Format	TD406

# X.CVE

- X.CVE is a literal copy of CVE Compatibility Requirements from the CVE Web Site
  - **Changes to CVE Compatibility Requirements will be reflected as updates to X.CVE**
  - **The CVE Editorial Board retains control of CVE**
- X.CVE will put CVE in a more “recognized” standards body versus “The MITRE Corporation” without taking control of the content or the requirements on CVE usage from the CVE Editorial Board





SCAP 1.1 uses the following specifications:

- Extensible Configuration Checklist Description Format (XCCDF) 1.1.4, a language for authoring security checklists/benchmarks and for reporting results of checklist evaluation [QUI08]
- Open Vulnerability and Assessment Language (OVAL) 5.6, a language for representing system configuration information, assessing machine state, and reporting assessment results
- Open Checklist Interactive Language (OCIL) 2.0, a language for representing security checks that requires human feedback
- Common Platform Enumeration (CPE) 2.2, a nomenclature and dictionary of hardware, operating systems, and applications [BUT09]
- Common Configuration Enumeration (CCE) 5, a nomenclature and configurations
- Common Vulnerabilities and Exposures (CVE), a nomenclature and software flaws<sup>9</sup>
- Common Vulnerability Scoring System (CVSS) 2.0, an open specification for severity of software flaw vulnerabilities [MEL07].

**NIST**  
National Institute of  
Standards and Technology  
U.S. Department of Commerce

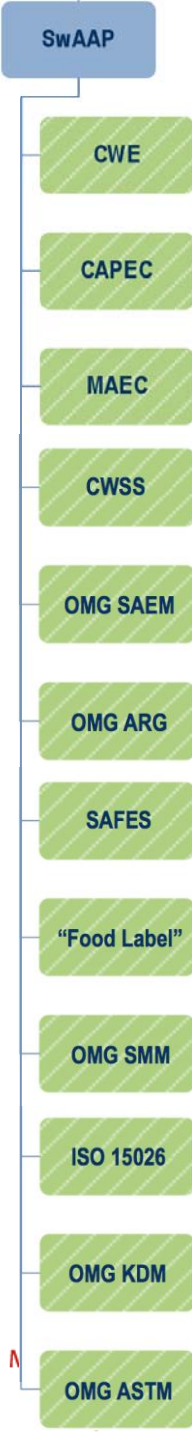
Special Publication 800-126  
Revision 1 (DRAFT)

## **The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.1 (DRAFT)**

Recommendations of the National Institute  
of Standards and Technology

Stephen Quinn  
David Waltermire  
Christopher Johnson  
Karen Scarfone  
John Banghart



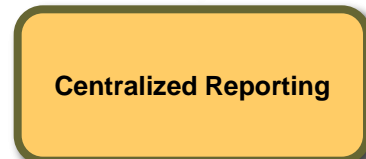
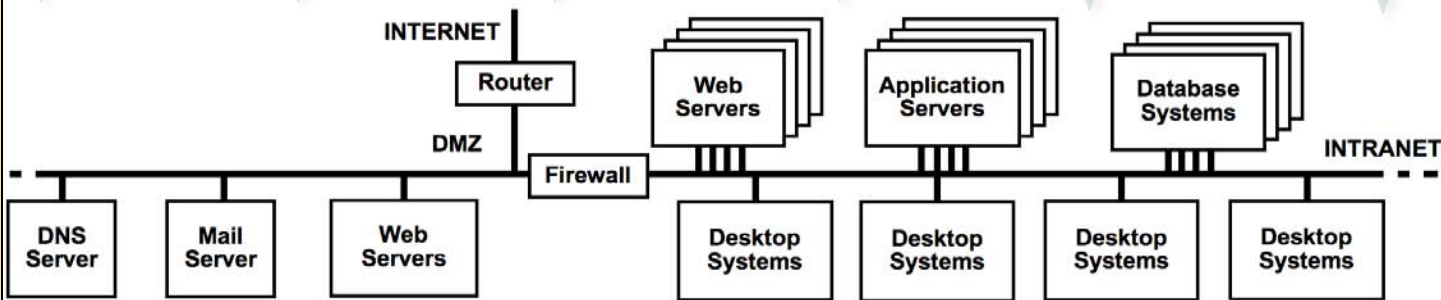
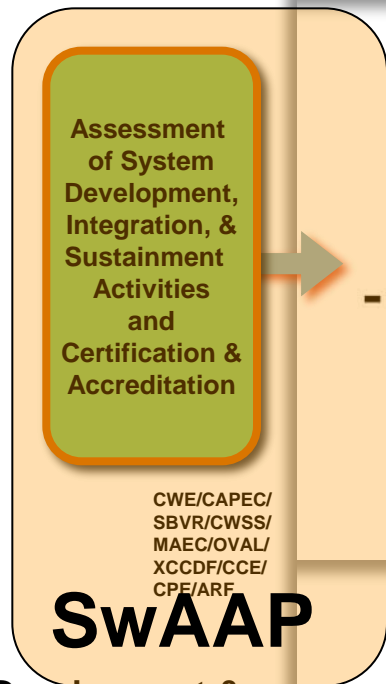
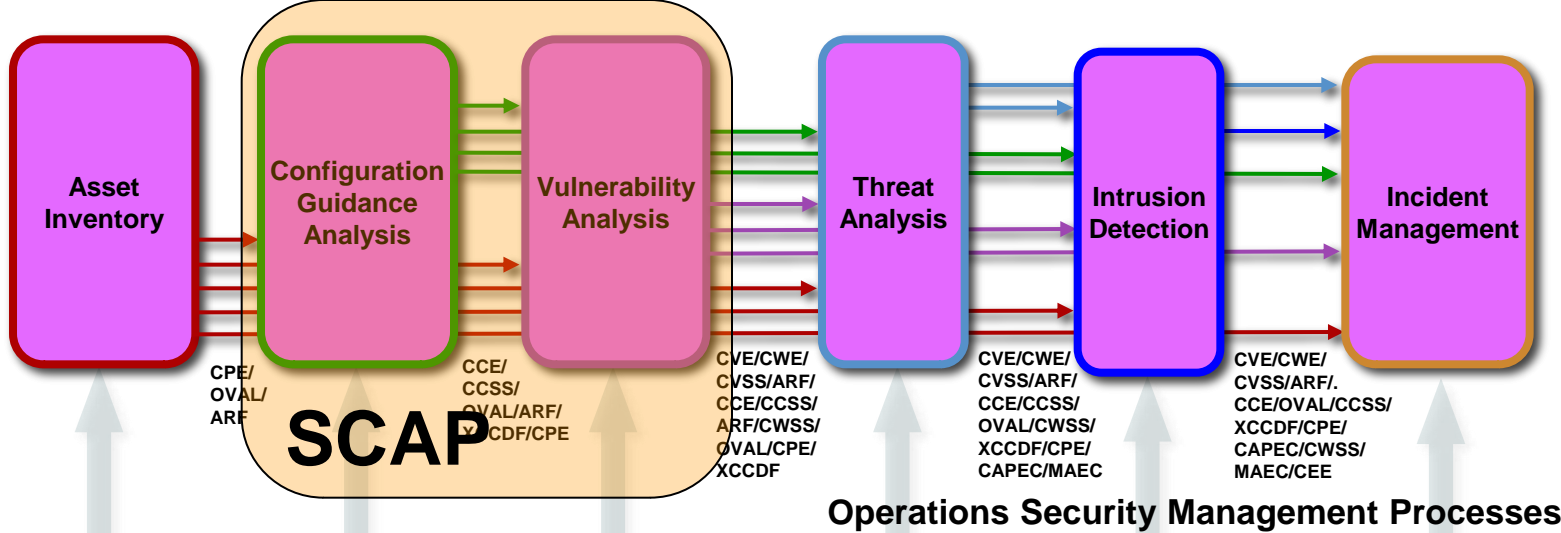
A vertical diagram on the left side of the slide shows the SwAAP framework. At the top is a blue box labeled 'SwAAP'. Below it is a vertical line with 13 green boxes branching off to the right. The boxes are labeled: CWE, CAPEC, MAEC, CWSS, OMG SAEM, OMG ARG, SAFES, "Food Label", OMG SMM, ISO 15026, OMG KDM, and OMG ASTM. The last three boxes (OMG KDM, ISO 15026, and OMG ASTM) have a red diagonal line through them.

# • Software Assurance Automation Protocol (**SwAAP**)

- **For measuring & enumerating software weaknesses and the assurance cases.**

Common Weakness Enumeration (**CWE**),  
Common Attack Pattern Enumeration & Classification (**CAPEC**),  
Malware Attribute Enumeration & Characterization (**MAEC**),  
Common Weakness Scoring System (**CWSS**),  
OMG Software Assurance Evidence Metamodel (**OMG SAEM**),  
OMG Argumentation Metamodel (**OMG ARG**),  
Software Assurance Findings Expression Schema (**SAFES**),  
NIST SAMATE's "Food Label",  
OMG Structured Metrics Metamodel (**OMG SMM**),  
ISO "Assurance Case" 15026 (**ISO 15026**),  
OMG Knowledge Discovery Metamodel (**OMG KDM**),  
OMG Abstract Syntax Tree Metamodel (**OMG ASTM**)

- **plus SCAP to capture "accredited" system CPEs and CCE settings?**
- **OVAL checks for capturing "finger print" of software applications to address supply-chain risk measurement?**



**Enterprise IT Asset Management**

# “Other” Automation Protocols (“O”AP)

- **Event Management Automation Protocol (EMAP)**
  - For reporting of security events. Common Event Expression (CEE), Malware Attribute Enumeration & Characterization (MAEC), and Common Attack Pattern Enumeration & Classification (CAPEC).
- **Enterprise Remediation Automation Protocol (ERAP)**
  - For automated remediation of mis-configuration & missing patches. Common Remediation Enumeration (CRE), Extended Remediation Information (ERI), Open Vulnerability and Assessment Language (OVAL), Common Platform Enumeration (CPE), and Common Configuration Enumeration (CCE).
- **Enterprise Compliance Automation Protocol (ECAP)**
  - For reporting configuration compliance. Asset Reporting Format (ARF), Open Checklist Reporting Language (OCRL), etc.
- **Enterprise System Information Protocol (ESIP)**
  - For reporting of asset inventory information. Common Platform Enumeration (CPE), etc.

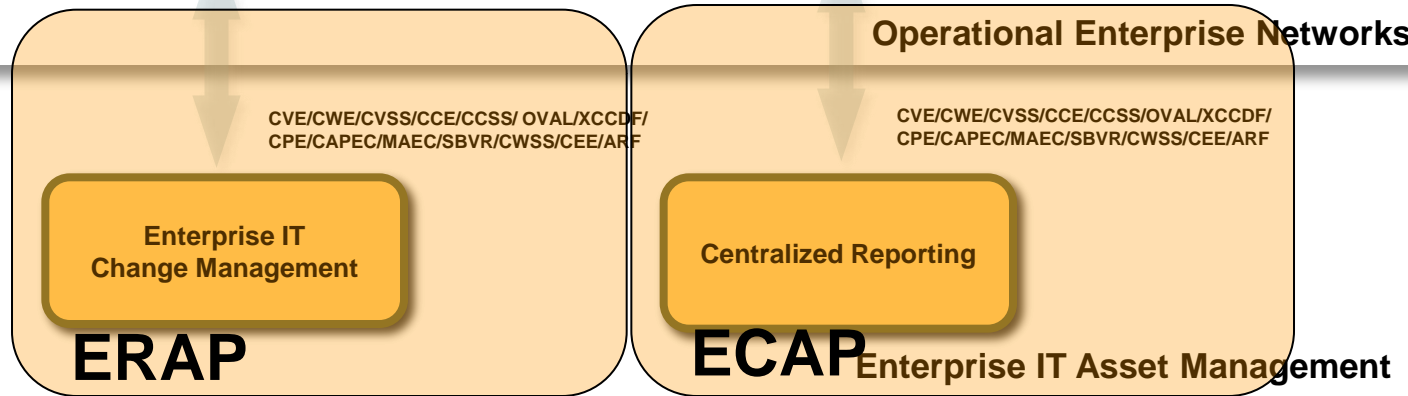
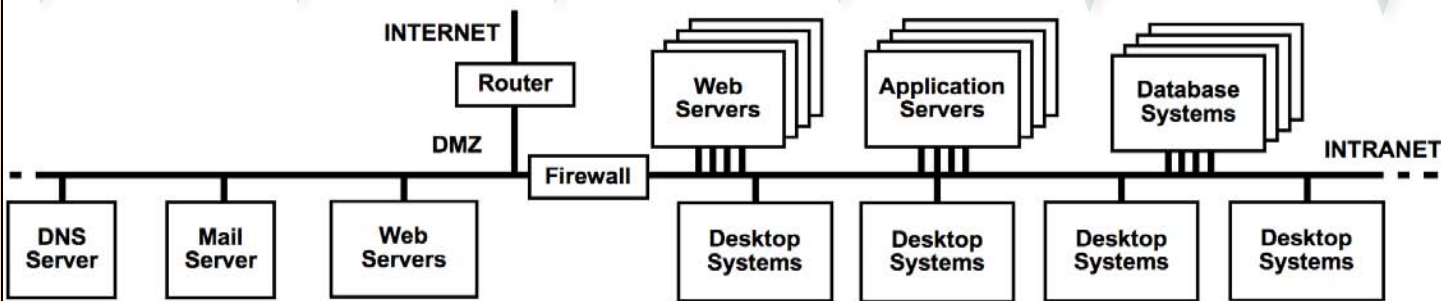
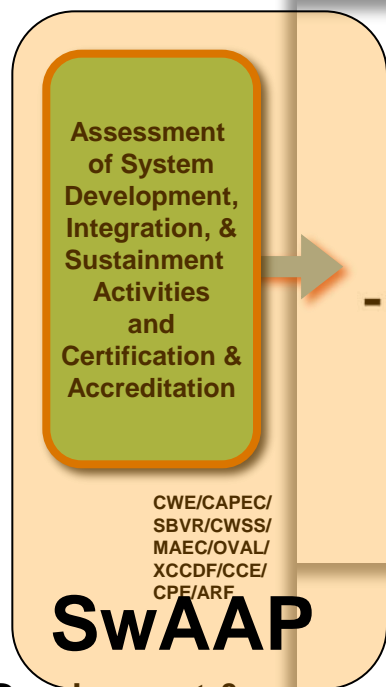
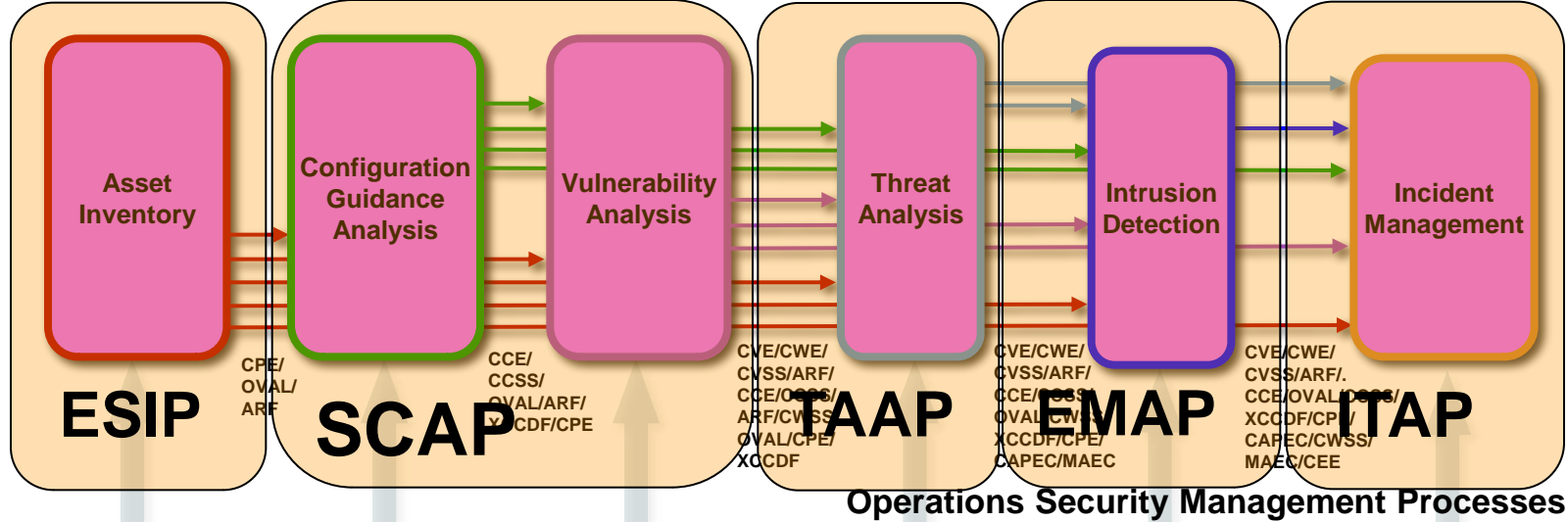
# “Other” Automation Protocols (“O”AP)

- **Incident Tracking and Assessment Protocol (ITAP)**

- For tracking, reporting, managing and sharing incident information. Open Vulnerability and Assessment Language (OVAL), Common Platform Enumeration (CPE), Common Configuration Enumeration (CCE), Common Vulnerabilities and Exposures (CVE), Common Vulnerability Scoring System (CVSS), Malware Attribute Enumeration & Characterization (MAEC), Common Attack Pattern Enumeration & Classification (CAPEC), Common Weakness Enumeration (CWE), Common Event Expression (CEE), Incident Object Description Exchange Format (IODEF), National Information Exchange Model (NIEM), and Cybersecurity Information Exchange Format (CYBEX).

- **Threat Analysis Automation Protocol (TAAP)**

- For reporting and sharing structured threat information. Malware Attribute Enumeration & Characterization (MAEC), Common Attack Pattern Enumeration & Classification (CAPEC), Common Platform Enumeration (CPE), Common Weakness Enumeration (CWE), Open Vulnerability and Assessment Language (OVAL), Common Configuration Enumeration (CCE), and Common Vulnerabilities and Exposures (CVE).





A blue-tinted photograph of a large manta ray swimming over a school of smaller fish. The manta ray is the central focus, with its long, curved tail and wide pectoral fins visible. The background is a deep blue, and the overall scene is serene and underwater.

# Questions?

[ramartin@mitre.org](mailto:ramartin@mitre.org)